出力系列のタスク指向初期化を用いた編集に基づくニューラル機械翻訳

大矢 一穂[1,a)]　吉永 直樹[2,b)]

概要：非自己回帰モデルは処理の並列化により高速な生成が可能だが，自己回帰モデルと比べて出力の品質が低下することが知られている．これに対し，出力系列に対して繰り返し編集操作を行う編集ベースの非自己回帰モデルは，推論速度は犠牲になるものの，高品質の出力が可能である．本論文では，編集に基づく非自己回帰モデルを用いた機械翻訳に焦点を当て，編集に基づく非自己回帰モデルの精度・推論速度の向上のために，トークン単位の翻訳を用いた出力系列の初期化手法を提案する．WAT2017 と WMT14 を用いた英日・英独翻訳実験を通じて，提案手法の有効性と今後の課題について議論を行う．

1. はじめに

Transformer [1] に基づく自己回帰（Autoregressive; AR）モデルは，高品質な文章を生成できるため，機械翻訳サービスやチャットボットといった，様々なテキスト生成サービスで主流となっている．一方で，AR モデルは出力長の二乗に比例した生成時間が必要となる．そのため，同時翻訳アプリや対話システムのような，瞬時に出力することが求められるサービスや，毎日大量のアクセスを受けるような大規模なテキスト生成サービスにおいて，AR モデルの生成時間の長さが問題となる場合がある．

この問題に対し，Gu ら [2] は，すべての目的言語トークンを同時に予測する非自己回帰（Non-autoregressive; NAR）モデルを提案した．NAR モデルは，処理の並列化により AR モデルよりも高速な生成が可能である．その一方で NAR モデルは出力するトークン間の依存関係を考慮することが難しく，品質や精度の面で AR モデルに劣ってしまう．これに対し，編集ベースの NAR モデル [3], [4], [5] は，並列的な編集操作を繰り返すことで生成文の品質を向上させる工夫を行う手法である．**図 1** の上の例は，編集ベースの NAR モデルによる文生成の様子である．空の系列に対し，挿入操作と再配置操作を繰り返すことによって文生成を行っている．近年では，空の系列から編集操作を始める代わりに，適切な初期出力系列に対して編集操作を適用することで，少ない編集操作で生成を行う手法が研究されている [6], [7].

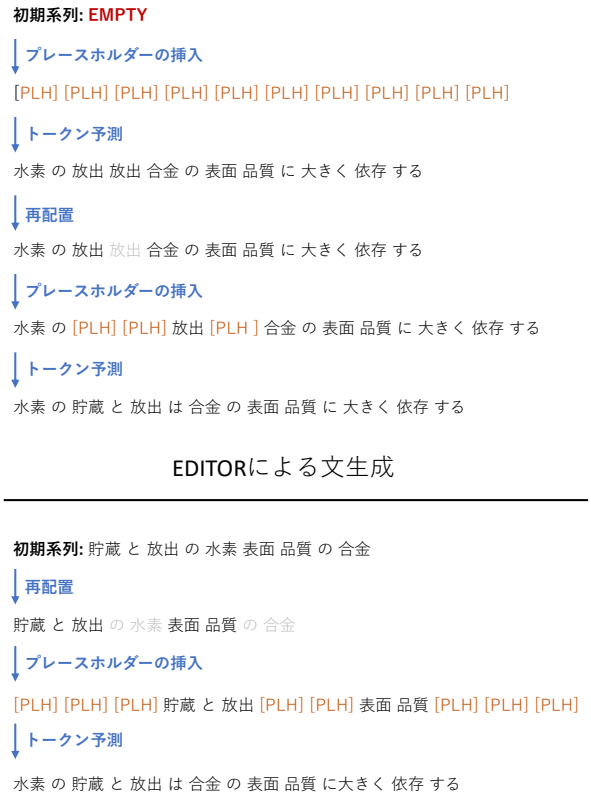

図 1　EDITOR（上）及び提案手法（下）による文生成の様子

本研究では，編集ベースの NAR モデルにおいて，新たな出力系列の初期化手法として，トークン単位の翻訳を行う手法を提案する（図 1 の下）．提案手法では，入力系列

1　東京大学大学院 情報理工学系研究科
2　東京大学 生産技術研究所
a)　ioya@tkl.iis.u-tokyo.ac.jp
b)　ynaga@iis.u-tokyo.ac.jp

の各トークンについて，学習データ中の最も相関の高いトークンにトークン単位で翻訳を行い，初期系列として用いる．トークン単位翻訳の際は，相関係数に基づく対訳表を作成し，さらにフィルタリングにより低品質な対訳関係を除去する．学習では，初期系列を用いた生成を行うために，ベースとする NAR モデルのノイズを含む出力を復元するタスクと，初期系列から出力を生成するタスクを組み合わせたマルチタスク学習を行う．

実験では, 編集ベースの NAR モデルである EDITOR に適用し, WAT2017 の ASPEC 日英コーパス [8] と WMT14 [9] を用いた英日・英独翻訳による提案手法の評価を行った．評価の結果，提案手法により EDITOR の推論速度の向上と編集操作回数の削減には成功した一方で，翻訳精度は EDITOR に劣ることが分かった．

2. 関連研究

本節では，非自己回帰モデルに基づくテキスト生成手法について説明する．Gu ら [2] は，系列長に関わらずすべての目的言語トークンの予測を同時に行う非自己回帰（NAR）モデルを提案している．彼らは Fertility という概念を導入し，原言語の各トークンに対応する目的言語のトークン数の予測を行い，予測した系列長の出力を 1 ステップで生成した．実験では，自己回帰（AR）モデルの 15 倍の推論速度向上を達成する一方で，同じトークンの繰り返しなど，NAR モデルの抱える出力系列の品質が低いという課題が明らかとなった．

以降では，まず，前述の NAR モデルの出力品質の改善手法として，編集に基づく NAR モデルを説明する．次に，編集ベースの NAR モデルの推論速度を改善するべく，出力系列の初期化を行う手法について述べる．

2.1 編集ベースの NAR モデル

NAR モデルの課題に対し，出力系列に対し並列的な編集操作を繰り返す，編集ベースの NAR モデルが研究されている [3], [4], [5]．編集ベースの NAR モデルは，純粋な NAR モデルのような 1 ステップでのデコードは行わず，出力系列に対し，トークン単位で並列に編集操作を繰り返し適用する．その結果，推論速度は落ちるものの純粋な NAR モデルよりも高品質な出力系列を生成することを可能にした．Stern ら [3] は，トークンの挿入操作を繰り返すことによって出力系列の生成を行う Insertion Transformer を提案した．Insertion Transformer では，隣接する各トークン間にそれぞれ最大で 1 トークンずつ挿入することを繰り返すことで，出力系列長 n に対し $\log_2 n$ 回程度のステップで出力系列の生成が可能である．また，Gu ら [4] は，挿入に加えて削除操作を導入し，1 組のトークン間に複数トークンの挿入を可能にした手法である，Levenshtein Transformer を提案した．Levenshtein Transformer では，トークンの削除，プレースホルダの挿入，各プレースホルダに対するトークン予測の 3 つの操作を繰り返し適用することで出力系列の生成を行った．

さらに Xu ら [5] は，Levenshtein Transformer をベースにしつつ，削除操作の代わりに再配置操作を行う EDITOR を提案した．再配置操作は，トークンの削除と同時にトークンの位置の入れ替えを行う操作であり，編集操作の過程で，トークン順の修正が可能であるという特徴がある．そのため，連続する 2 トークンが逆順になっている場合など，Levenshtein Transformer では削除と挿入の 2 ステップが必要な操作を，EDITOR では 1 回の再配置操作で行うことができる．

これらの手法では，1 ステップで生成を行う純粋な NAR モデルと比較して，高品質の出力を行える一方で，複数ステップにわたって生成を行うため，推論時間が増加するという問題がある．

2.2 編集ベース NAR モデルにおける出力系列の初期化

編集ベースの NAR モデルが空の系列から編集操作を始めるのに対し，最終的な出力系列に類似した系列から編集をする手法が研究されている．Xu ら [7] は，Levenshtein Transformer をベースにしながら，検索による出力系列の初期化を行う TM-LevT を提案した．例えば，仏文 “Un chat dort” から英文 “A cat is sleeping” への翻訳をする際，編集距離をもとに学習データを検索し，類似した仏文である “Un chat mange” と対応する英文である “The cat is eating” を見つけ，“The cat is eating” から編集操作により出力系列の生成を行う．このように目的とする出力系列に類似した系列から編集操作を行うことで，必要な編集操作の回数が少なくなり，より高精度な出力生成を可能にした．また Niwa ら [6] は，原言語文をベクトル化し，そのコサイン類似度をもとに初期文を検索する手法を提案した．彼女らの手法は，JRC-Acquis を用いた英独翻訳実験において，NAR のベースラインである Levenshtein Transformer を上回る翻訳精度と推論速度を達成した．一方で，WNT14 [9] を用いた実験では，ベースラインよりも編集操作の回数が増加し，むしろ推論速度が低下してしまうなど，データセットに依存して性能にぶれがあることを報告している．この速度低下について，彼女らは，データセット中に類似した文が少ないことが原因であると分析している．

これらの研究が，文単位で検索を行い初期系列とするのに対し，本研究では，入力系列をトークン単位で翻訳することによって初期系列の生成を行う．

3. 予備知識: EDITOR

本節では，本研究で基盤として用いる編集ベースの NAR モデルである Edit-Based TransfOrmer with Repositioning (EDITOR) [5] を説明する．本モデルは，Transformer

encoder-decoder モデルのアーキテクチャを用いている．

3.1 編集操作による出力系列の生成

編集ベースの NAR モデルは，初期出力系列 $\boldsymbol{y}^0$ に対し編集操作を繰り返し適用することで出力系列を洗練させていく．k 番目の繰り返しでは，出力系列 $\boldsymbol{y}^{k-1} = (y_1, y_2, ..., y_n)$ に対して，編集操作 a^{k+1} を行うことで，新たな出力系列 $\boldsymbol{y}^{k+1}$ を得る．特に EDITOR は，再配置操作及び，プレースホルダの挿入とトークン予測の 2 つからなるトークン挿入を交互に繰り返すことで出力系列の生成を行う．

再配置操作は，出力系列中のトークンの削除及び並べ替えを行う操作である．系列 $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ の各位置 i に対してインデクス $r \in [0, n]$ を予測し，$r > 0$ のとき，i 番目の位置にトークン y_r を配置する．一方で $r = 0$ のときは，位置 i にトークンを配置せず，これが実質的な削除操作にあたる．実際の再配置は，位置 i におけるデコーダの出力 $\boldsymbol{h}_i$ と系列 $\boldsymbol{y}$ の各トークンの埋め込み $\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n$ からインデクス r の確率分布を計算する分類器を学習し，これを用いて行う．

$$\pi_{rps}(r|i, \boldsymbol{y}) = \mathrm{softmax}(\boldsymbol{h}_i \cdot [\boldsymbol{b}, \boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n]) \quad (1)$$

ただし，ベクトル $\boldsymbol{b} \in \mathbb{R}^{d_{model}}$ は $r = 0$ に対応しておりトークンの削除を表している．

プレースホルダの挿入操作は，出力系列 $\boldsymbol{y}$ 中の隣接するトークン (y_i, y_{i+1}) に対して，トークン間に新たに挿入するトークンの数 $p \in [0, K_{max}]$ を予測する．

$$\pi_{plh}(p|i, \boldsymbol{y}) = \mathrm{softmax}([\boldsymbol{h}_i; \boldsymbol{h}_{i+1}] \cdot \boldsymbol{W}^{plh}) \quad (2)$$

ただし，; はベクトルの連結を表す．そして，予測した数のプレースホルダ [PLH] をトークン間に挿入する．

トークン予測操作では，系列中の y_i =[PLH] であるすべての i に対して，挿入するトークン t の予測を行う．

$$\pi_{tok}(t|i, \boldsymbol{y}) = \mathrm{softmax}(\boldsymbol{h}_i \cdot \boldsymbol{W}^{tok}) \quad (3)$$

3.2 学習

まず参照文 $\boldsymbol{y}^*$ に対し，ランダムにトークンを削除しシャッフルを行った $\boldsymbol{y}'$ を生成する．挿入操作の学習時，$\boldsymbol{y}'$ および $\boldsymbol{y}'$ にモデルの再配置操作を適用した系列の混合である $\boldsymbol{y}_{ins}$ を用意し，$\boldsymbol{y}_{ins}$ を $\boldsymbol{y}^*$ に復元する最適なトークン挿入を行うように，挿入操作の学習を行う．

再配置操作の学習の際は，$\boldsymbol{y}'$ および $\boldsymbol{y}'$ にモデルの挿入操作を適用した系列の混合である $\boldsymbol{y}_{rps}$ を用意し，$\boldsymbol{y}_{rps}$ を $\boldsymbol{y}^*$ に復元する最適な再配置操作を行うようにモデルの再配置操作の学習を行う．

3.3 推論

推論時は，空の系列である初期出力系列 $\boldsymbol{y}^0$ に対し，再配置（r），プレースホルダの挿入 (p)，トークン予測（t）の 3 つの操作を一連の工程として，この工程を $(a^1, a^2, ...) = (r^1, p^1, t^1; r^2, p^2, t^2; ...)$ のように繰り返し適用することで出力系列の生成を行う．

以下の 2 つの少なくとも片方が満たされたとき，推論を終了し，その時点での出力系列を最終的な出力系列とする．

- 一連の工程を行った前後で出力系列が変化していない場合．つまりこれは，出力系列に対しモデルが挿入操作も再配置も行わなかったときか，もしくは再配置とトークン挿入が相殺してループにはまったときである．
- 繰り返しの回数があらかじめ定めた上限回数に達した場合．

4. 提案手法

編集ベースの NAR モデルに適切な初期出力系列を用いることで，編集操作の回数が減ることによる推論時間の短縮や翻訳精度の向上が見込める．しかし既存の研究では類似文検索かつ，特定の言語対（仏英，英独）の研究にとどまっており，また類似文検索による手法では，データセット中の類似文の多さによって性能が左右されることが示唆されている．

そこで本研究では，トークン単位の翻訳によって初期系列を生成する手法を提案し，NAR モデルである EDITOR に適用する．

4.1 対訳表の生成

まず原言語のトークンと目的言語のトークンのすべての組み合わせについて，学習データの対訳文のペアの中に出現しているかどうかの二値データとして，相関係数を計算する．原言語のトークン w_s と目的言語のトークン w_t の相関係数を計算する場合，学習データ中の対訳文対 (s_i, t_i) に対し，

$$x_i = \begin{cases} 1 & (w_s \in s_i) \\ 0 & (w_s \notin s_i) \end{cases} \quad (4)$$

$$y_i = \begin{cases} 1 & (w_t \in t_i) \\ 0 & (w_t \notin t_i) \end{cases} \quad (5)$$

を計算する．得られた $X = (x_1, x_2, ...)$ と $Y = (y_1, y_2, ...)$ を用いて相関係数 r_{xy} を計算し，それを w_s と w_t の相関係数とする．

その後，原言語のトークンそれぞれに対し，最も相関係数の高い目的言語のトークンを選択し，対訳表の生成を行う．

4.2 対訳表に対するフィルタリング

生成した対訳表の中には初期系列として有用ではない対訳関係も含まれている．例えば，WAT2017 の英日対訳デー

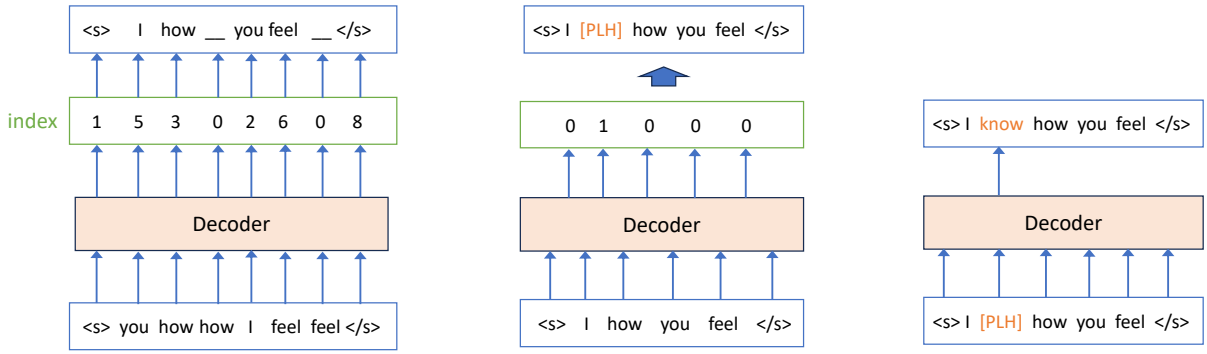

図 2 EDITOR の編集操作

English token	Japanese token (highest correlation)	Correlation
from	から	0.574
Quantum	量子	0.206
quantum	量子	0.754
wire	ワイヤ	0.578
mature	成熟	0.589
such	などの	0.573
new	新しい	0.668
density	密度	0.686

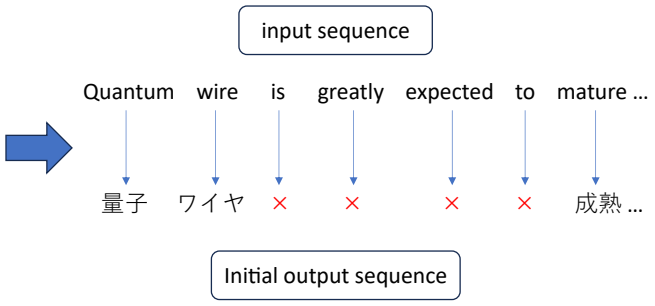

図 3 提案手法による初期系列生成

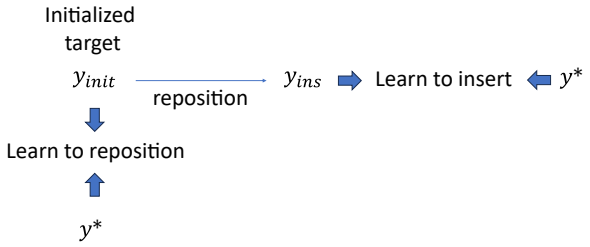

図 4 初期出力系列を用いた学習のデータフロー

タで生成した対訳表では，英語トークン “The” に対して日本語トークンの「は」が対応している．この 2 つのトークンの相関係数は 0.100 であったが，他に相関係数の高いトークンが存在しなかったため，日本語の文の中で出現頻度の高い「は」が対応するトークンとして選択されている．また，英語トークン “for” に対応している日本語トークンとして「のための」が選択されているが，実際に “for” が出現した英文のうち，対応する日本語の文で「のための」が出現したのは約 6%であり，“for” に対する翻訳として「のための」というトークンはほとんど使われていない．この対訳は，訳としては妥当であるものの，編集操作によって置換される可能性が高く，必ずしも編集ベースの NAR モデルの初期系列を構成するのに適当でない可能性がある．

そこで，対訳表に対するフィルタを設けて初期系列に有用ではない対訳関係の除去を行う．原言語のトークン w_s に対し目的言語のトークン w_t が対応しているとき，以下のようなトークンを対訳表から除外する．

- 相関係数 $(w_s, w_t) < 0.2$
- $\dfrac{w_s \text{と } w_t \text{が含まれる文対の数}}{w_s \text{ が含まれる原言語の文の数}} < \dfrac{1}{3}$

4.3 初期出力系列の生成

入力系列の各トークンについて，対応している目的言語のトークンに置き換え，それを初期出力系列とする．対訳表にないトークンは置き換えを行わずスキップする．

図 3 は，対訳表を用いた初期出力系列の生成の様子である．対応しているトークン同士は簡易的な訳語の関係になっており，初期出力系列は，入力系列をトークン単位で翻訳した系列に相当している．

4.4 モデルの学習

EDITOR の学習（3.2 節）では，ノイズが付与された参照文を復元する学習を行う．しかし，ノイズが付与された参照文と初期系列は傾向が異なるため，EDITOR の学習方法では，初期系列から文を生成する過程を学習することが難しい．そこで学習時は，EDITOR のノイズを含む参照文の復元タスクと，対訳表によって生成した初期出力系列から参照文を生成するタスクでマルチタスク学習を行う（**図 4**）．初期系列を用いる学習では，まず，初期系列 $\boldsymbol{y}_{init}$ に対し，参照文 $\boldsymbol{y}^*$ を生成する最適な再配置操作を行うようにモデルの再配置操作の学習を行う．さらに，$\boldsymbol{y}_{init}$ に再

表 1 翻訳タスクにおける精度と推論速度の評価結果.
提案手法は括弧書きで初期系列の平均生成時間を併記している.

	モデル	知識蒸留	BLEU	COMET	平均生成時間 (ms)		繰り返し回数
En-Ja	AR (6-6)		42.89	90.10	158.1		22.36
	AR (11-1)		41.29	88.98	65.5		21.71
	EDITOR		40.24	89.03	84.4		3.15
	EDITOR + initial sequence		39.24	88.42	80.5	(4.3×10^{-3})	2.66
	EDITOR	✓	40.98	89.15	76.0		2.81
	EDITOR + initial sequence	✓	40.73	88.76	69.5	(4.6×10^{-3})	2.19
En-De	AR (6-6)		26.16	83.07	196.9		27.65
	AR (11-1)		25.20	78.74	79.0		27.05
	EDITOR		22.90	78.00	87.2		3.33
	EDITOR + initial sequence		19.67	75.48	81.6	(4.0×10^{-3})	2.75
	EDITOR	✓	24.15	79.85	68.4		2.53
	EDITOR + initial sequence	✓	23.68	78.86	63.8	(4.0×10^{-3})	2.00

配置操作を適用した系列を $\boldsymbol{y}_{ins}$ として，$\boldsymbol{y}_{ins}$ を $\boldsymbol{y}^*$ に復元する最適な挿入操作を行うように挿入操作の学習を行う．学習時は，各データについて，EDITOR の学習と初期出力系列を用いた学習を 1/2 ずつになるように確率的に選択し，選択した学習を行う．

5. 実験設定

提案手法による効果を検証するために，言語対の異なる 2 つの機械翻訳データセットを用いて実験を行う．

5.1 データセット

英日翻訳タスクのデータセットとして，WAT2017 の ASPEC 日英コーパス [8] を使用した．英独翻訳タスクには，WMT14 [9] を使用した．

5.2 翻訳モデル

実験では，以下のモデルの比較を行う．

AR AR のベースラインとして，Transformer [1] のベースサイズモデル（6 層のエンコーダ，6 層のデコーダ）を使用した．また，Kasai ら [10] の研究に倣い，推論速度の優れた AR モデルとして 11 層のエンコーダと 1 層のデコーダからなる Transformer モデルも使用した．

NAR NAR モデルとして，EDITOR および EDITOR に提案手法を適用したモデルを用いた．モデルのハイパーパラメータは Transformer のベースサイズモデルと同じものを使用する．

また，先行研究 [2], [4] では，NAR モデルに AR の教師モデルを用いた知識蒸留 [11] を行うことで，翻訳精度が向上することが報告されている．この場合の知識蒸留とは，学習データの参照文を，AR の教師モデルの出力した翻訳文に置き換えて学習を行うことである．本研究では，Transformer のベースサイズモデルを用いて知識蒸留を行った場合と行わなかった場合それぞれについて実験を行った．

5.3 評価

翻訳精度の評価には BLEU [12] と COMET [13] を用いた．BLEU を計算する際は，SacreBLEU で計算を行った [14]. 推論速度の評価には，バッチサイズ 1 でデコーディングした際の 1 文あたりにかかった平均時間を計算した．また，提案した初期系列を用いたモデルに対しては，初期系列生成にかかる時間を別途計測した．実験はシードを変えてそれぞれ 3 回行い，平均したスコアを報告する．

6. 実験結果

表 1 に，実験結果を示す．BLEU 及び COMET を用いた翻訳精度の評価において，提案した初期系列を用いたモデルは，NAR ベースラインモデルの EDITOR を下回った．一方で生成時間は，EDUTIR から 5%から 9%程度減少し，生成に必要な繰り返し回数も減少している．つまり，提案手法により翻訳精度は低下するものの推論速度と繰り返し回数は改善している．

また，EDITOR と提案手法はどちらもベースサイズ AR モデル（6-6）の翻訳精度を下回っている．対して推論速度では NAR モデルが AR モデル（6-6）を大きく上回っている．デコーダの層を減らした AR モデル（11-1）は AR モデル（6-6）に翻訳精度は劣るものの推論速度が向上しており，英日翻訳実験では提案手法よりも高速な推論を行っている．一方で英独翻訳実験では，提案手法が AR モデル（11-1）の推論速度を上回っている．これは，今回行った英独翻訳実験が，英日翻訳実験に比べて平均出力長が長いからであると考えられる．

表 2 は，英日翻訳実験における，EDITOR 及び提案手法の文生成の様子を示している．例 1 では，初期系列を用

表 2 EDITOR 及び提案手法の英日翻訳における翻訳例.
ここではプレースホルダの挿入とトークン予測をまとめて挿入操作と表記している．また，EDITOR が初めに行う空の系列に対する再配置操作（再配置 1）は省略している.

例 1	
原言語文	A scattering of oxygen gas molecule from graphite surface is simulated by molecular dynamics method.
目的言語文	黒鉛表面からの酸素ガス分子の散乱を分子動力学法でシミュレーションした.
	EDITOR
初期系列	empty
挿入 1	グラファイト表面からの酸素ガス分子の分子動力学をシミュレートした.
再配置 2	グラファイト表面からの酸素ガス分子のをシミュレートした.
挿入 2	分子動力学（グラファイト表面からの酸素ガス分子の散乱を分子動力学法によりシミュレートした.
再配置 3	グラファイト表面からの酸素ガス分子の散乱を分子動力学法によりシミュレートした.
	EDITOR + initial sequence
初期系列	散乱の酸素ガス分子から黒鉛表面シミュレート分子動力学
再配置 1	黒鉛の酸素ガス散乱シミュレート
挿入 1	黒鉛表面からの酸素ガス分子の散乱を分子動力学法によりシミュレートした.
例 2	
原言語文	The United States intends to promote the international and domestic competitive power and superiority as a supreme proposition by close cooperation of government, industry, and university.
目的言語文	米国は自国の国際競争力優位を至上命題として政府・産業界・大学の密接な協力によって推進しようとしている.
	EDITOR
初期系列	empty
挿入 1	米国は官官産産のの緊連携による提案提案として国際の競争力競争力優位優位を推進するようとしている.
再配置 2	米国は官産の緊連携によるとして国際競争力と優位を推進するようとしている.
挿入 2	米国は官産学の緊密な連携による前提提案として，国際国内国内競争力と優位性を推進するようとしている.
再配置 3	米国は官産学の緊密な連携による前提提案として，国際国内競争力と優位性を推進するようとしている.
挿入 3	米国は，官産学大学の緊密な連携による前提提案として，国際・国内の競争力と優位性を推進するようとしている.
再配置 4	米国は，官産学大学の緊密な連携による前提提案として，国際・国内の競争力と優位性を推進するようとしている.
挿入 4	米国は，官産学，大学の緊密な連携による前提提案として，国際・国内の競争力と優位性を推進するようとしている.
	EDITOR + initial sequence
初期系列	米国を促進する国際と国内競合と優位協力の政府，産業，と大学
再配置 1	政府政府大学大学の協力，国際と優位
挿入 1	政府・政府・大学・大学の緊密な協力により，前提として国際国内国内競争力と優位性推進ようようようとしている.
再配置 2	政府・・の緊密な協力により，前提として国際国内競争力と優位性ようとしている.
挿入 2	政府・産業・大学の緊密な協力により，前提として，国際・国内の競争力と優位性を推進しようとしている.
例 3	
原言語文	This paper discusses the mechanism of the heat return reaction.
目的言語文	熱戻り反応の機構を議論した
	EDITOR
初期系列	empty
挿入 1	熱リターン反応の機構を論じた.
	EDITOR + initial sequence
初期系列	を論じた機構の熱反応
再配置 1	empty
挿入 1	熱戻反応の機構を論じた.
再配置 2	熱戻反応の機構を論じた.
挿入 2	熱戻り反応の機構を論じた.

いることで，必要な編集操作の回数が半分に減少している．また例 2 では，提案手法により編集操作の回数は減少しているものの，「米国は」という内容が出力系列から抜け落ちてしまっている．例 3 では，EDITOR が挿入操作 1 回で生成を行っているのに対し，提案手法は，初めの再配置操作でトークンをすべて削除してしまい，EDITOR より必要な操作回数が多くなってしまっている．

7. おわりに

本研究では，編集ベースの非自己回帰モデルに対し，トークン単位の翻訳による初期系列生成を提案した．実験の結果，提案手法により，推論速度及び必要な繰り返し回数の改善には成功した一方で，翻訳精度が低下してしまった．今後は，翻訳精度改善のため，対訳表の生成手法や学習方法の見直しを行っていきたい．また，今回使用した EDITOR 以外の NAR モデルに提案手法を適用することも検討したい．その上で，検索ベースの系列初期化手法との比較も行っていきたい．

謝辞 本研究は，東京大学生産技術研究所特別研究経費および JSPS 科研費 JP21H03494 の支援を受けたものである．

参考文献

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems* (Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R., eds.), Vol. 30, Curran Associates, Inc., (online), available from ⟨https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf⟩ (2017).

[2] Gu, J., Bradbury, J., Xiong, C., Li, V. O. and Socher, R.: Non-Autoregressive Neural Machine Translation, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=B1l8BtlCb⟩ (2018).

[3] Stern, M., Chan, W., Kiros, J. and Uszkoreit, J.: Insertion Transformer: Flexible Sequence Generation via Insertion Operations, *Proceedings of the 36th International Conference on Machine Learning* (Chaudhuri, K. and Salakhutdinov, R., eds.), Proceedings of Machine Learning Research, Vol. 97, PMLR, pp. 5976–5985 (online), available from ⟨https://proceedings.mlr.press/v97/stern19a.html⟩ (2019).

[4] Gu, J., Wang, C. and Zhao, J.: Levenshtein Transformer, *Advances in Neural Information Processing Systems* (Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. and Garnett, R., eds.), Vol. 32, Curran Associates, Inc., (online), available from ⟨https://proceedings.neurips.cc/paper_files/paper/2019/file/675f9820626f5bc0afb47b57890b466e-Paper.pdf⟩ (2019).

[5] Xu, W. and Carpuat, M.: EDITOR: An Edit-Based Transformer with Repositioning for Neural Machine Translation with Soft Lexical Constraints, *Transactions of the Association for Computational Linguistics*, Vol. 9, pp. 311–328 (online), DOI: 10.1162/tacl_a_00368 (2021).

[6] Niwa, A., Takase, S. and Okazaki, N.: Nearest Neighbor Non-autoregressive Text Generation, *Journal of Information Processing*, Vol. 31, pp. 344–352 (online), DOI: 10.2197/ipsjjip.31.344 (2023).

[7] Xu, J., Crego, J. and Yvon, F.: Integrating Translation Memories into Non-Autoregressive Machine Translation, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (Vlachos, A. and Augenstein, I., eds.), Dubrovnik, Croatia, Association for Computational Linguistics, pp. 1326–1338 (online), DOI: 10.18653/v1/2023.eacl-main.96 (2023).

[8] Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S. and Isahara, H.: ASPEC: Asian Scientific Paper Excerpt Corpus, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J. and Piperidis, S., eds.), Portorož, Slovenia, European Language Resources Association (ELRA), pp. 2204–2208 (online), available from ⟨https://aclanthology.org/L16-1350⟩ (2016).

[9] Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L. and Tamchyna, A.: Findings of the 2014 Workshop on Statistical Machine Translation, *Proceedings of the Ninth Workshop on Statistical Machine Translation* (Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M. and Specia, L., eds.), Baltimore, Maryland, USA, Association for Computational Linguistics, pp. 12–58 (online), DOI: 10.3115/v1/W14-3302 (2014).

[10] Kasai, J., Pappas, N., Peng, H., Cross, J. and Smith, N.: Deep Encoder, Shallow Decoder: Reevaluating Non-autoregressive Machine Translation, *International Conference on Learning Representations*, (online), available from ⟨https://openreview.net/forum?id=KpfasTaLUpq⟩ (2021).

[11] Kim, Y. and Rush, A. M.: Sequence-Level Knowledge Distillation, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (Su, J., Duh, K. and Carreras, X., eds.), Austin, Texas, Association for Computational Linguistics, pp. 1317–1327 (online), DOI: 10.18653/v1/D16-1139 (2016).

[12] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: a Method for Automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (Isabelle, P., Charniak, E. and Lin, D., eds.), Philadelphia, Pennsylvania, USA, Association for Computational Linguistics, pp. 311–318 (online), DOI: 10.3115/1073083.1073135 (2002).

[13] Rei, R., Stewart, C., Farinha, A. C. and Lavie, A.: COMET: A Neural Framework for MT Evaluation, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Webber, B., Cohn, T., He, Y. and Liu, Y., eds.), Online, Association for Computational Linguistics, pp. 2685–2702 (online), DOI: 10.18653/v1/2020.emnlp-main.213 (2020).

[14] Post, M.: A Call for Clarity in Reporting BLEU Scores, *Proceedings of the Third Conference on Machine Trans-*

表 A·1　データセットの詳細

dataset	train	valid	test
WAT2017	2,000,000	1,792	1,812
WMT14	3,961,179	3,000	3,003

表 A·2　学習の詳細

hyper-parameters	value
label smoothing	0.1
number of max tokens	21,600
dropout rate	0.3
optimizer	adam
adam β	(0.9, 0.98)
learning rate	5×10^{-4}
warmup lr	1×10^{-7}
warmup updates	10,000
max updates	300,000

lation: Research Papers (Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Monz, C., Negri, M., Névéol, A., Neves, M., Post, M., Specia, L., Turchi, M. and Verspoor, K., eds.), Brussels, Belgium, Association for Computational Linguistics, pp. 186–191 (online), DOI: 10.18653/v1/W18-6319 (2018).

[15] Kudo, T. and Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Blanco, E. and Lu, W., eds.), Brussels, Belgium, Association for Computational Linguistics, pp. 66–71 (online), DOI: 10.18653/v1/D18-2012 (2018).

付　　録

A.1　データセットの詳細

表 A·1 にデータセットの詳細を示す．WAT2017 については, 配布されているサブワード分割済みのデータ[*1]を用いた．サブワード分割では，語彙サイズ 16384 になるように SentencePiece [15] を使用して行われている．WMT14 については, fairseq で配布されているスクリプト[*2]を用いて前処理とサブワード化を行った．検証データには newstest2013 を，テストデータには newstest2014 を使用した．

A.2　学習・評価の詳細

表 A·2 に学習の詳細を示す．実装は, 公開されている EDITOR のソースコード[*3]を用いて行った．モデルのチェックポイントは検証データの BLEU スコアを元に選択した．また, Stern ら [3] や Gu ら [4] と同様に, NAR モデルの推論時, プレースホルダを挿入しないことに対するペナルティを設けることで, 過度に文が短くなることを防ぐ EOS penalty の項を導入した．EOS penalty の項を [1.5, 3.0] の間で 0.5 刻みで変化させ, 検証データの BLEU スコアが最も高い値を選択した．検証およびテストに用いた SacreBLEU の signature は，英日翻訳は “BLEU|nrefs:1|case:mixed|eff:no|tok:ja-mecab-0.996-IPA|smooth:exp|version:2.4.3”，英独翻訳では “BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.3” である．

[*1] https://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2017/snmt/index.html
[*2] https://github.com/facebookresearch/fairseq/tree/main/examples/translation#wmt14-english-to-german-convolutional
[*3] https://github.com/weijia-xu/fairseq-editor